**INTRODUCCION**

El siguiente trabajo final tiene como principal objetivo demostrar el correcto uso de herramientas den IA con un enfoque en el campo del análisis de datos, parte fundamental en el campo de la Inteligencia Artificial. Por lo cual abordaremos la manipulación, procesamiento y visualización de información utilizando un enfoque práctico.

El desarrollo del trabajo se centrará en el consumo de dos datasets con orígenes distintos: uno provendrá de un archivo en formato CSV y la otra fuente vendrá de una base de datos SQLITE. La lectura y manejo inicial de estos conjuntos de datos se hará a través de la librería Pandas, reconocida por su eficiencia en la estructuración y manipulación de datos.

La fase de análisis culminará en la generación de varios gráficos, los cuales serán creadas utilizando librerías especializadas como Matplotlib, Bokeh o PyWalker. Todo el desarrollo del trabajo, desde la ingesta hasta la visualización, se llevará a cabo dentro de la plataforma JupyterLab (entorno local), facilitando un entorno interactivo para la exploración y desarrollo. El resultado de este proceso.

**MARCO TEORICO**

**Pandas**

Es una librería de código abierto que proporciona estructuras de datos y herramientas de análisis de datos de alto rendimiento

**Características principales:**

* Posee una sintaxis intuitiva y fácil de aprender, lo que permite a los usuarios manipular y analizar datos rápidamente.
* Está optimizado para manejar grandes volúmenes de datos de manera eficiente, lo que lo hace ideal para proyectos de análisis de datos a gran escala.
* Cuenta con documentación detallada y una gran comunidad de usuarios, lo que proporciona abundantes recursos y soporte para resolver problemas y aprender nuevas técnicas.
* GitHub - Pandas: El repositorio oficial de Pandas en GitHub es donde se desarrolla la librería. Aquí puedes encontrar el código fuente, reportar problemas y contribuir al desarrollo.

**Usos comunes:**

* Limpieza y transformación de datos para análisis.
* Estadísticas descriptivas y visualización de datos.

**SQLITE**

Es una biblioteca de base de datos SQL embebida, ligera y autónoma que permite almacenar y gestionar datos de manera eficiente sin necesidad de un servidor externo. Es muy popular en aplicaciones de escritorio, móviles (Android/iOS) y sistemas embebidos debido a su bajo consumo de recursos.

**Características principales:**

* Sin servidor: Los datos se guardan en un solo archivo (.db o .sqlite).
* Multiplataforma: Compatible con Windows, Linux, macOS, Android, iOS, etc.
* Soporte SQL completo: Incluye transacciones, triggers, vistas y JOINS.
* Lenguajes soportados: Python, C/C++, Java, JavaScript (Node.js), PHP, etc.
* Licencia pública: Libre de uso incluso en proyectos comerciales.

**Matplotlib**

Es una biblioteca de visualización de datos en Python, diseñada para crear gráficos estáticos, animados e interactivos de alta calidad.

**Características principales:**

* Matplotlib está diseñado para ser intuitivo y flexible, lo que lo hace ideal tanto para principiantes como para usuarios avanzados.
* Sintaxis clara y minimalista.
* No requiere configuración compleja para gráficos básicos.
* Ideal para prototipado rápido.
* Matplotlib cuenta con una documentación detallada y bien organizada, que incluye tutoriales, ejemplos y referencias de API.
* Tiene una gran comunidad de usuarios y contribuidores, lo que facilita encontrar soluciones a problemas comunes.
* Stack Overflow: Más de 300,000 preguntas etiquetadas con matplotlib.
* GitHub: Repositorio oficial con issues y discusiones: matplotlib/matplotlib.
* Foros y Redes: Subreddits como r/Python y grupos de Discord.

**Bokeh**

Es una biblioteca de visualización interactiva en Python diseñada para crear gráficos atractivos y altamente interactivos que pueden ser mostrados en los navegadores web.

**Características principales:**

* Es una de las bibliotecas más potentes para visualización interactiva en Python, especialmente diseñada para la web
* Permite crear gráficos interactivos con pocas líneas de código.
* Se pueden añadir sliders, botones y menús desplegables con facilidad.
* Funciona directamente con DataFrames
* Tiene una documentación clara y bien organizada, con ejemplos prácticos y tutoriales paso a paso.
* En Stack Overflow más de 15,000 preguntas etiquetadas con bokeh.
* En GitHub cuenta con repositorio oficial con contribuciones activas (bokeh/bokeh).
* Comunidades dedicadas a visualización en Python.

**Usos comunes:**

* Dashboards Web: Monitoreo de datos en tiempo real
* Exploración de Datos: Filtrado interactivo con widgets
* Aplicaciones Científicas: Visualización de simulaciones o modelos 3D.
* Informes Empresariales: Gráficos incrustados en páginas web o aplicaciones internas.

**PyWalker**

Es una biblioteca de Python diseñada para simplificar el análisis exploratorio de datos (EDA) mediante una interfaz interactiva similar a Tableau. Se integra con populares herramientas como Pandas y Jupyter Notebook, permitiendo la generación de visualizaciones dinámicas (gráficos, tablas, dashboards) con arrastrar y soltar (*drag-and-drop*), sin la necesidad de escribir código complejo.

**Características principales:**

* Interfaz intuitiva: Panel lateral interactivo para seleccionar variables, tipos de gráficos y filtros.
* Soporte para DataFrames: Funciona directamente con Pandas, Polars o conexiones a bases de datos (SQL).
* Visualizaciones personalizables: Histogramas, dispersión, mapas de calor, opciones muy variadas.
* Integración con Jupyter: Ideal para entornos de notebooks (Colab, Jupyter Lab).

**DESCRIPCION DEL DATASET**

El siguiente dataset utilizado para realizar el trabajo final es resultado de consumir la siguiente API: <https://archive-api.open-meteo.com/v1/archive> esta URI nos permite tener información climatológica de cualquier parte del mundo dicha información la guardamos en un archivo .csv llamado “DS\_Mediciones\_Clima.csv”, la página climática Open-Meteo ofrece acceso a una gran cantidad de datos meteorológicos, basándose en conjuntos de datos de reanálisis. Este reanálisis combina observaciones de diferentes fuentes como estaciones meteorológicas, aviones, boyas, radares, satélites y utilizan modelos matemáticos para estimar valores donde faltan datos, permitiendo así obtener información detallada incluso en lugares sin estaciones cercanas, como áreas rurales o el océano.

El dataset generado posee 17903 registros con 11 columnas de datos, la principal información que presente este conjunto de datos es: código de ciudad, fecha, temperatura, velocidad del viento, niveles de neblina, nivel de lluvia, presión, temperatura del suelo, altura, humedad relativa.

**PASOS REALIZADOS**

**1.- Creación de la data**

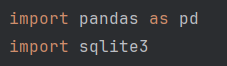
* Consumimos la API climatológica y generamos el archivo.csv “DS\_Mediciones\_Clima.csv” el cual contiene 17903 registros.
* Creamos la base de datos mediante el archivo base\_datos.py
* Desde el cmd del sistema nos dirigimos a la carpeta que contiene el proyecto y levantamos la base de datos con el siguiente comando: ‘’pyhton base\_datos.py”
* Toda esta información debe estar guardada en la carpeta que contiene el proyecto

**2.- Levantamiento del proyecto**

* Abrimos Git Bash y nos dirigimos a la carpeta que alberga todo el proyecto y ejecutamos el siguiente comando: “jupyter-lab” y se nos abrirá la interfaz web del Jupyter NotBook
* Antes de empezar con la codificación debemos verificar que estén las carpetas de /Base y /Data en estas esta la información que se utilizar para la codificación.
* Ya en la interfaz vamos crear un nuevo notebook donde empezaremos con toda la codificación del proyecto el nombre del archivo es el siguiente: “MainNotebook.ipynb”

**3.- Codificación**

* Importamos las librerías: pandas y sqlite3

****

* Leemos el archivo .cvs y creamos el df mediciones donde almacenaremos al Info del .csv



* Nos conectamos con la base y creamos el puntero cursor



****

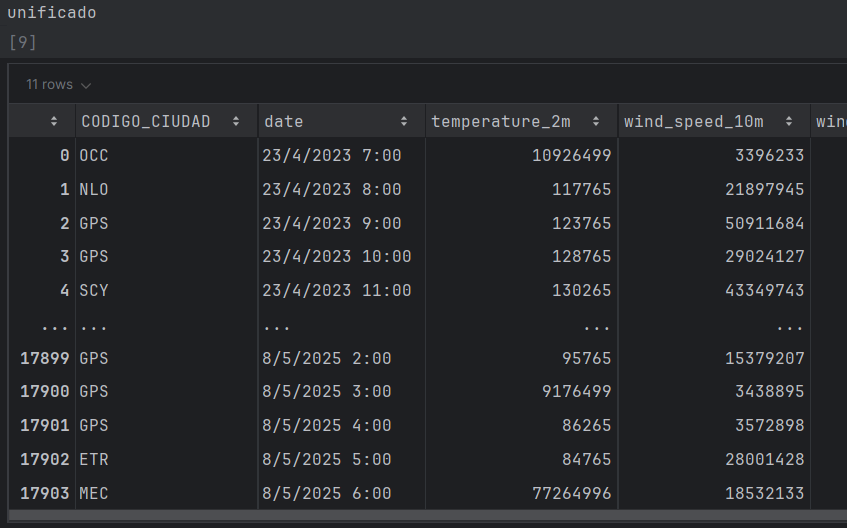
* Creamos un df llamado clima donde se almacenará la información de la base

****

* Ya lista la Info ahora la unificaremos mediante un merge con el siguiente comando

****

* El df unificado será ahora nuestra principal información para proceder con la visualización grafica de la información.

****

**4.- Gráficos**

**DESCRIPCION DE LAS VISUALIZACIONES GENERADAS**

**CONCLUSIONES**

**BIBLIOGRAFIA**

Mucci, T., & Finio, M. (2025, 8 abril). Casos de uso de la IA generativa. *Casos de Uso Pandas*. <https://www.ibm.com/mx-es/think/topics/generative-ai-use-cases>

DataScientest. (2023, 30 octubre). *Pandas : La biblioteca de Python dedicada a la Data Science*. <https://datascientest.com/es/pandas-python>

EBIS Business Techschool. (2024, 12 agosto). *Pandas en Python: Qué es y Cómo Instalarlo (Guía 2025)*. EBIS Education. <https://www.ebiseducation.com/pandas-en-python-que-es-y-como-instalarlo-guiaScikit-learn>

*SQLite Home page*. (s. f.). <https://www.sqlite.org/>

*SQLite: Documentation*. (s. f.). <https://www.sqlite.org/src/doc/trunk/README.md>

Anthony\_Manotoa. (2025, 27 febrero). *Qué es y cómo usar Matplotlib: la guía definitiva (instalación y uso)*. https://platzi.com. <https://platzi.com/blog/matplotlib/>

*Using Matplotlib — Matplotlib 3.10.1 documentation*. (s. f.). <https://matplotlib.org/stable/users/index.html>

*Bokeh documentation*. (s. f.). Bokeh. <https://docs.bokeh.org/en/latest/index.html>

*User guide*. (s. f.). Bokeh. <https://docs.bokeh.org/en/latest/docs/user_guide.html>

Kanaries. (2023). \*PyWalker: Turn your DataFrame into a Tableau-style UI\*. GitHub. <https://github.com/Kanaries/pywalker>

Autor, A. (2024). \*Interactive Data Visualization in Python with PyWalker\*. Towards Data Science.